3D Point Cloud Reconstruction from a Single 4D Light Field Image

Helia Farhood*[a], Stuart Perry[a], Eva Cheng[b], Juno Kim[c]
[a]Perceptual Imaging Laboratory, School of Electrical and Data Engineering, University of Technology Sydney, Sydney, Australia; [b]School of Professional Practice and Leadership, University of Technology Sydney, Sydney, Australia; [c]School of Optometry and Vision Science, University of New South Wales, Sydney, Australia

ABSTRACT

Obtaining accurate and noise-free three-dimensional (3D) reconstructions from real world scenes has grown in importance in recent decades. In this paper, we propose a novel strategy for the reconstruction of a 3D point cloud of an object from a single 4D light field (LF) image based on the transformation of point-plane correspondences. Considering a 4D LF image as an input, we first estimate the depth map using point correspondences between sub-aperture images. We then apply histogram equalization and histogram stretching to enhance the separation between depth planes. The main aim of this step is to increase the distance between adjacent depth layers and to enhance the depth map. We then detect edge contours of the original image using fast canny edge detection, and combine linearly the result with that of the previous steps. Following this combination, by transforming the point-plane correspondence, we can obtain the 3D structure of the point cloud. The proposed method avoids feature extraction, segmentation and the extraction of occlusion masks required by other methods, and due to this, our method can reliably mitigate noise. We tested our method with synthetic and real world image databases. To verify the accuracy of our method, we compared our results with two different state-of-the-art algorithms. In this way, we used the LOD (Level of Detail) to compare the number of points needed to describe an object. The results showed that our method had the highest level of detail compared to other existing methods.

Keywords: 3D Reconstruction, Plenoptic image, Light field, 3D point cloud, Single 4D Light Field Image

1. INTRODUCTION

The issue of acquiring noise-less and complete 3D point clouds is of paramount importance to support advances in virtual, augmented reality and 3D printing. There is a significant demand for obtaining high quality 3D point clouds scanned from real objects for 3D rendering, computer graphics, 2D view extraction from 3D data, virtual reality, and object deformation [1]. A 3D point cloud can be obtained from various methods. Many existing methodologies for reconstruction of 3D models are based on either Structure-From-Motion (SFM) or Dense Multi-View 3D Reconstruction (DMVR). However, these methods need multiple captures from different angles and significant user interaction when using an ordinary camera. For this reason, recent research has focused on the development of new strategies using less costly devices while continuing to minimize complexity [2].

Given the status quo, one very effective solution is the creation of a 3D model from one single image. However as a single conventional image supplies limited information about the 3D nature of objects or scenes, accurate estimation of 3D geometry from a single image is difficult to achieve [3]. Nevertheless, light field images with the ability to capture rich 3D information with a single capture can be an effective solution to this problem. The introduction of light field cameras (a type of plenoptic camera) has helped to reveal new solutions and insights into a wide variety of traditional computer vision and image processing problems. Light field cameras capture rich information about the intensity, color and direction of light, and can be used to estimate a depth map or 3D point cloud from a single captured frame. Light field camera technology has the potential to create 3D point cloud reconstructions in circumstances where standard multi-capture techniques can fail such as dynamic scenes or objects with complicated material appearance. The unique features of light field images, such as the capture of light rays from multiple directions, provides the ability to reconstruct 2D images at different focal planes. This feature can also aid in depth map reconstruction [2]. A light field image can have multiple representations. However, two LF representation formats are more common for computer vision and image processing problems; the lenslet format and 4D LF format. In this work, we used the 4D format that in our experience is more reliable

*helia.farhood@student.uts.edu.au

for depth map estimation. For the reconstruction of a 3D point cloud of an object, we develop new method which is based on the transformation of the point-plane correspondences. The input of our system is a 4D LF image which can be captured by light field cameras, such as the Lytro [4] or created synthetically. As a first step we estimate a sparse version of the depth map using sub-aperture image matching. As having densely sampled depth map is essential for creating a 3D point cloud, we enhance the depth map substantially by applying histogram equalization and histogram stretching followed by adding edge detection information from the original image. This kind of enhancement after estimation of depth map is one of the significant contributions of this work. These steps also can increase the distance between adjacent depth layers. Therefore, we combined the result of stretched depth map with canny edge detection results linearly and then transform the point-plane correspondence for acquiring a 3D point cloud. Figure 1 shows the main block diagram of our proposed method.

The remainder of the paper is arranged as follows: Section 2 presents prior work relating to reconstruction of 3D point cloud, and estimation of the depth map. Section 3 discusses briefly plenoptic camera characteristics. In Section 4 our proposed method is described. Performance on real-world and synthetic image databases are compared against other methods discussed in Section 5, followed by conclusive remarks and potential future directions in Section 6.

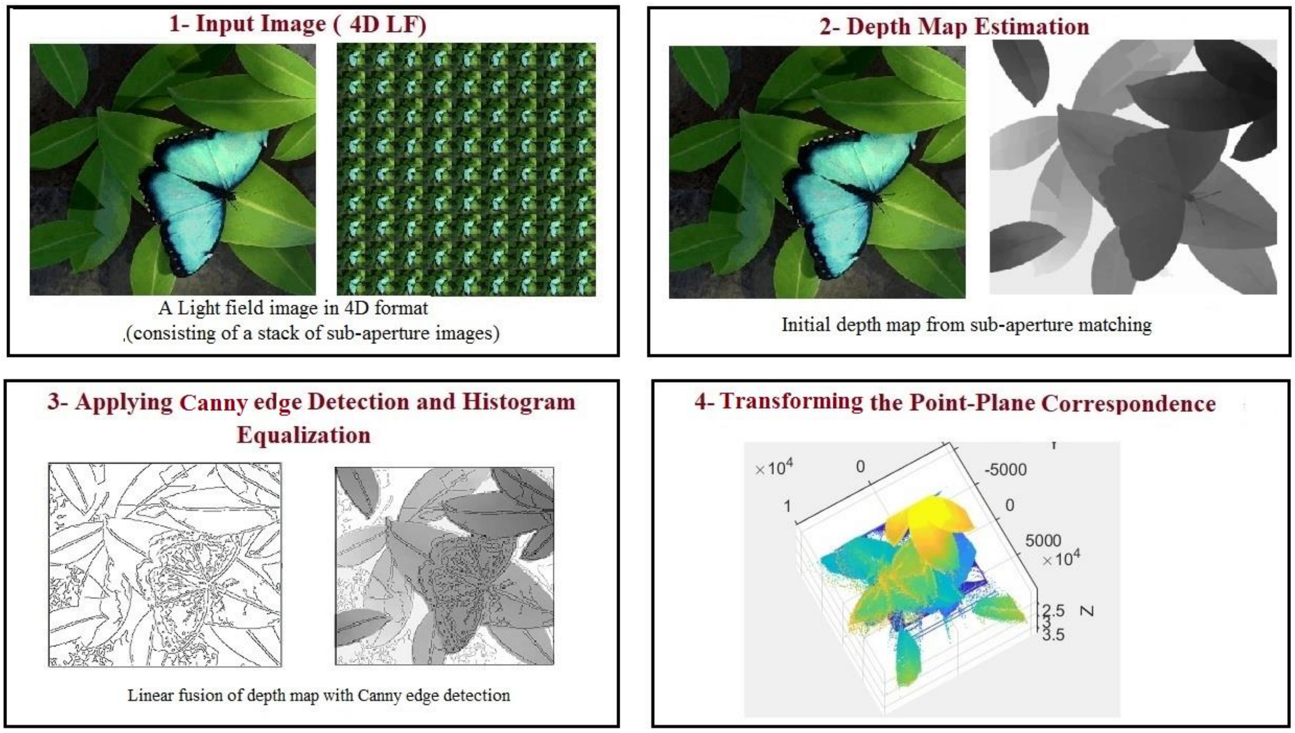
1- Input Image (4D LF)
A Light field image in 4D format
.(consisting of a stack of sub-aperture images)
2- Depth Map Estimation
Initial depth map from sub-aperture matching
3- Applying Canny edge Detection and Histogram Equalization
Linear fusion of depth map with Canny edge detection
4- Transforming the Point-Plane Correspondence
×10^4
0
-5000
1
0
5000
×10^4
2.5
3
3.5
Z

Figure 1. Overview of the proposed method. Our method consists of 4 main stages: 1. 4D format of light field image as input images, 2. Depth map estimation based on using point correspondences between sub-aperture images, 3. Canny edge detection results are combined linearly with the histogram stretched depth map, 4. Transforming the point plane correspondences to create a final 3D point cloud.

2. RELATED WORK

Existing 3D reconstruction approaches can be divided into three different groups: methods based on capturing multiple images, creating 3D point clouds based on Deep Learning and approaches based on 3D reconstruction from a single image.

2.1 3D reconstruction from capturing multiple images

In general, the reconstruction of 3D point clouds from multiple image captures of the same scene is a computationally expensive process and requires significant user interaction. One of the common methods in this field is Structure from Motion (SFM) which requires the capture of photos of a scene from all feasible angles around the object. Pezzuolo et al. [5] reconstructed three-dimensional volumes of rural buildings from groups of 2-D images by using SFM methods. Bae et al. [6] proposed an image-based modelling technique as a faster method for 3D reconstruction. For image capture, they utilized cameras on mobile devices. One of the benefits of using image-based modelling is the accessibility of texture information that can enable material recognition and 3D CAD model objects recognition [7, 8] . Pileun et al. [9] estimated the positions and orientations of the object by using Simultaneous Localization and Mapping (SLAM). The 2D localization information is utilized for creating 3D point clouds. This reduces the time of scanning and requires less effort for collecting accurate 3D point cloud data but still needs user interaction.

2.2 Creating 3D point clouds based on Deep Learning techniques

Recently, approaches based on deep learning have drawn attention for solving many computer vision problems. A wide variety of deep learning models have been developed to create 3D point clouds but most of them require images capturing an object with an uncluttered background and a fixed view point [10]. Current techniques have limited application to real world objects. Yang et al. [11] generated point cloud based on a specific deep model named PointFlow This model has the advantage of having two levels of continuous flows of normalizing the point cloud. The first level is for creating the shape and the second level is for distributing the points. For handling large scale 3D point clouds, Wang et al. [12] developed a method based on the feature description matrix (FDM) combining traditional feature extraction with a deep leaning approach. As deep learning solely is not efficient for creating a 3D point cloud, Vetrivel et al. [13] combined a convolutional neural networks approach with 3D features to improve results. Wang et al. [14] used deep learning for fast segmentation of 3D point clouds. They introduced a new framework called Similarity Group Proposal Network (SGPN). However this method is still not efficient for real-world objects.

2.3 3D reconstruction from a single image approaches

Creating 3D point clouds from a single image has received significant attention from the research community. However, 3D reconstruction from a single projection still has many problems and is a challenging topic. Mandika et al. [15] estimated 3D point clouds from a single input view by training an auto-encoder to learn a mapping from 2D input images to 3D point clouds. However this type of estimate is not very accurate and requires extensive training [16]. To overcome the drawbacks of estimation of 3D point clouds from a single capture, light field cameras have been proposed. Using light field images as an input can lead 3D point cloud estimates with low cost and complexity. Perra et al. [2] used light field images as an input image to determine depth maps of scenes and from that information estimate 3D point clouds. The depth maps that are automatically acquired from light field cameras has have some potential limitations for when dealing with real objects. To tackle this problem, we propose a novel algorithm for estimating the depth map based on the information inherent to light field images.

3. LIGHT FIELD CAMERA

A light field camera is a type of Plenoptic camera that with one image capture through an array of micro-lenses, can collect a wide variety of information about the color, intensity and direction of light in a scene. In contrast, in a traditional digital camera, the lack of data about the direction and intensity of light makes solving many problems of computer vision difficult. An early light field camera was developed in 2005 and was called the Lytro [4] and professional version of Lytro was introduced in 2014, called the Lytro Illum. Light field cameras have several applications such as post-capture refocusing, depth map estimation and illumination estimation. One of the significant features of light field camera is shown in Figure 2 which shows post-capture refocusing in two different focal planes. Refocusing allows for changing the focal plane to a different position post-capture. A light field can be considered as a vector function *I (u, v, s, t)* and two planes (lens plane and sensor plane) [17] where *u* and *v* are coordinates on the lenses plane and *s* and *t* are co-ordinates on the sensor plane as shown in Figure 2.

4. PROPOSED METHOD

We used a Lytro Illum camera in this work. After comparing different types of LF formats we decided to use the 4D LF format as an input image for more consistency and reliability.

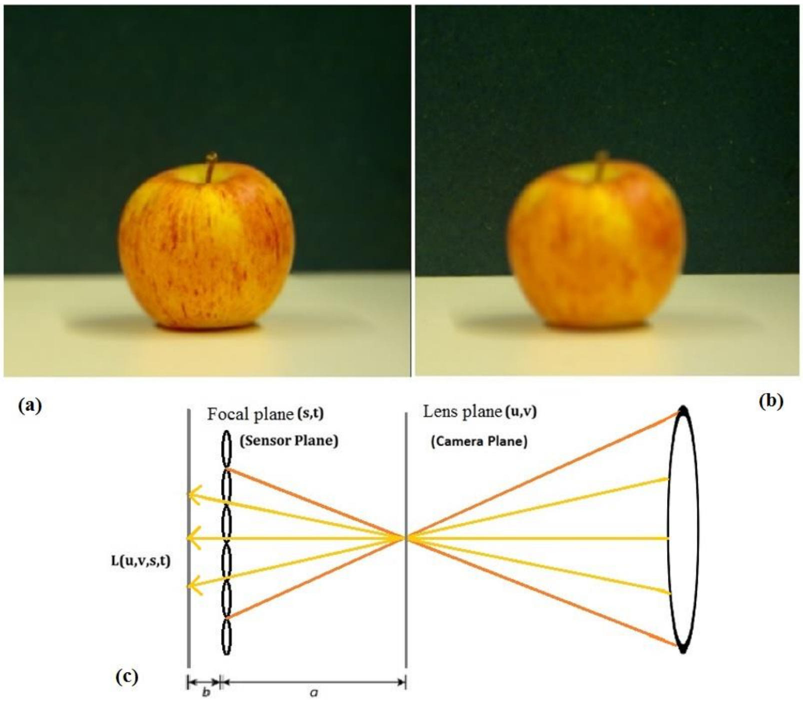

Figure 2. An illustration of the light field function and post-capture re-focusing. (a) Focus in the foreground. (b) Focus in the background. (c) A diagram of a light field camera setup where *a* is the distance from the camera plane to sensor plane and *b* is the distance between the sensor and the micro-lens array.

Our approach is based on transforming the point-plane correspondences and in the first step we estimate the depth map. Estimating an accurate depth map is essential to having a complete 3D point cloud. Following this step, we will enhance the depth map by histogram stretching and edge detection and then by transforming we will acquire the 3D point cloud.

4.1 Depth map Estimation

A light field image consists of an array of sub-aperture images. A sub-aperture image is defined by light rays entering the the focal (sensor) plane from one point in the lens plane. As we are utilizing the 4D LF format pixel coordinates of a LF image *I* are determined using the 4D parameters of (s, t, u, v). An initial depth map estimate is formed by correspondence matching between sub-aperture images.

For correspondence matching between sub-aperture images, the sum of gradients is first calculated. We begin by introducing P as a pixel in a non-central sub-aperture image and P_c as the corresponding pixel in the center view acquired by correspondence matching. For the correspondence matching we used the SIFT algorithm [18]. We can then define k as the label of the correspondence match and (s, t) are the coordinates of the sub-aperture image corresponding to P and (s_c,t_c) are the coordinates of the central sub-aperture image. For each correspondence the sum of gradients $GD(s,t,P,P_c)$ is acquired by a weighted sum of the gradients along the s and t dimensions:

$$GD(s,t,P,Pc) = GD(s,t,P,Pc) + \|I(P) - I(P_c)\| + \left\|I(P) - \overline{I(P)}\right\| \tag{1}$$

where $\overline{I(P)}$ is the median intensity value of the correspondences in s,t. A full-size map $G\acute{D}(s,t,u,v)$ is computed by allocating an array for each sub-aperture image of the same size as the sub-aperture images containing only zero values at each pixel location. For each correspondence, $G\acute{D}(s,t,u,v)$ is set to $GD(s,t,P,Pc)$. This results in a map of gradients for each sub-aperture image which contains the value of $GD(s,t,P,Pc)$ for locations where a correspondence was found and zeros elsewhere.

To obtain the final depth map based on correspondence matching we sum across all sub-aperture gradient maps.

$$DM(u,v) = \sum_{s,t} \acute{GD}(s,t,u,v) \tag{2}$$

For having a dense depth map we added defocus information as well to fill gaps between correspondences. We will save the depth map output as an 8-bit grayscale 2D image and for simplicity denote it as DM_{uv}.

4.2 Histogram equalization and Canny edge detection

After receiving the 2D depth map image from previous step, $DM(u,v)$, In order to increase the distance between adjacent depth layers we will enhance the depth map by histogram equalization. In the first step we are using histogram stretching to improve the separation between the depth planes.

$$\begin{aligned} m_{uv} &= \frac{DM_{max} - DM_{min}}{255} \\ b_{uv} &= 255 - m_{uv} * DM_{max} \\ HS_{uv} &= m_{uv} * DM_{uv} + b_{uv} \end{aligned} \tag{3}$$

where DM_{uv} is 2D depth map image with a minimum value denoted as DM_{min} and a maximum value denoted as DM_{max} and HS_{uv} is the histogram stretched depth map.

In the second step, canny edge detection is applied on the input image $I(u,v)$. The main aim of the Canny operator is to utilize the first derivative of a Gaussian in different directions as a filter of noise and then on the filtered image the maximum value of local gradient will be calculated to determine image edges [19].
For smoothing filter, the Canny filter uses a Gaussian filter as denoted below:

$$G(u,v) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{u^2+v^2}{2\pi\sigma^2}\right) \tag{4}$$

which is applied to the input image $I(u,v)$ by convolution.

$$\acute{I}(u,v) = G(u,v) * I(u,v) \tag{5}$$

After that for detecting the edges, the value of the local gradient and direction of image are calculated.

$$\begin{gathered} I_1(u,v) = (\acute{I}(u,v+1) - \acute{I}(u,v) + \acute{I}(u+1,v+1) - \acute{I}(u+1,v))/2 \\ I_2(u,v) = (\acute{I}(u,v) - \acute{I}(u+1,v) + \acute{I}(u,v+1) - \acute{I}(u+1,v+1))/2 \\ CA_{uv} = \sqrt{I_1(u,v)^2 + I_2(u,v)^2} \\ \theta(u,v) = \arctan(I_1(u,v) + I_2(u,v)) \end{gathered} \tag{6}$$

where $\theta(u,v)$ is the direction of gradient and CA_{uv} is the edge image.

In the third step, the histogram stretched image and the edge image are combined linearly (image fusion), and the result will be saved in T_{uv}.

$$T_{uv} = |HS_{uv} + CA_{uv}| \tag{7}$$

4.3 Creating 3D point cloud by transforming the point-plane correspondence

For estimation of 3D point cloud we need to estimate T_z (the *z* component of each point at position *u*,*v*) from which a point cloud can be created by transforming the point-plane.
For the points in the final point cloud we start by selecting $T_x = T_u$ and $T_y = T_v$, then for computing T_z:

$$D = b \ * fl * fc$$

$$T_z = \frac{D}{T_{xy} * fc * b * \max(T_{xy})} \tag{8}$$

which b is amount of the baseline, fl is the focal length and fc is the focus distance. fl is an intrinsic parameter and depends on the captured image. b and fc are extrinsic parameters and depend on the type of camera. The *x* and *y* coordinates of the point are then given by:

$$\acute{\mathrm{T}}_x = \frac{\mathrm{T}_x * T_z}{fc}$$

$$\acute{T}_y = \frac{\mathrm{T}_y * T_z}{fc} \tag{9}$$

where Se is the sensor size (mm). We then denote the estimation of 3D point cloud by T_{xyz} :

$$T_{xyz} = (\acute{T}_x \ , \acute{T}_y, T_z) \tag{10}$$

5. EXPERIMENTAL RESULTS

We evaluated our method with two different databases and compared the proposed method with two sate of art methods, as described in further detail below. For estimating the error of proposed method we used LOD (level of details) as shown in part 5.3.

5.1 Result on databases

We used two different databases; one synthetic database and one real world light field image database. For the synthetic database we utilized a database popularized by the research community, which was created by Wanner et al. [20]. For the dataset of real images we used a Lytro Illum to capture various images. For having better evaluation we captured images in different situations, including images with shading, low-texture, and challenging images such as very bright images and images with occluded pixels. Our method is tested in wide variety of images of which a sample of results on synthetic image is shown in Figure 1 and two other samples of real images captured ourselves are shown in Figure 3 and Figure 4.

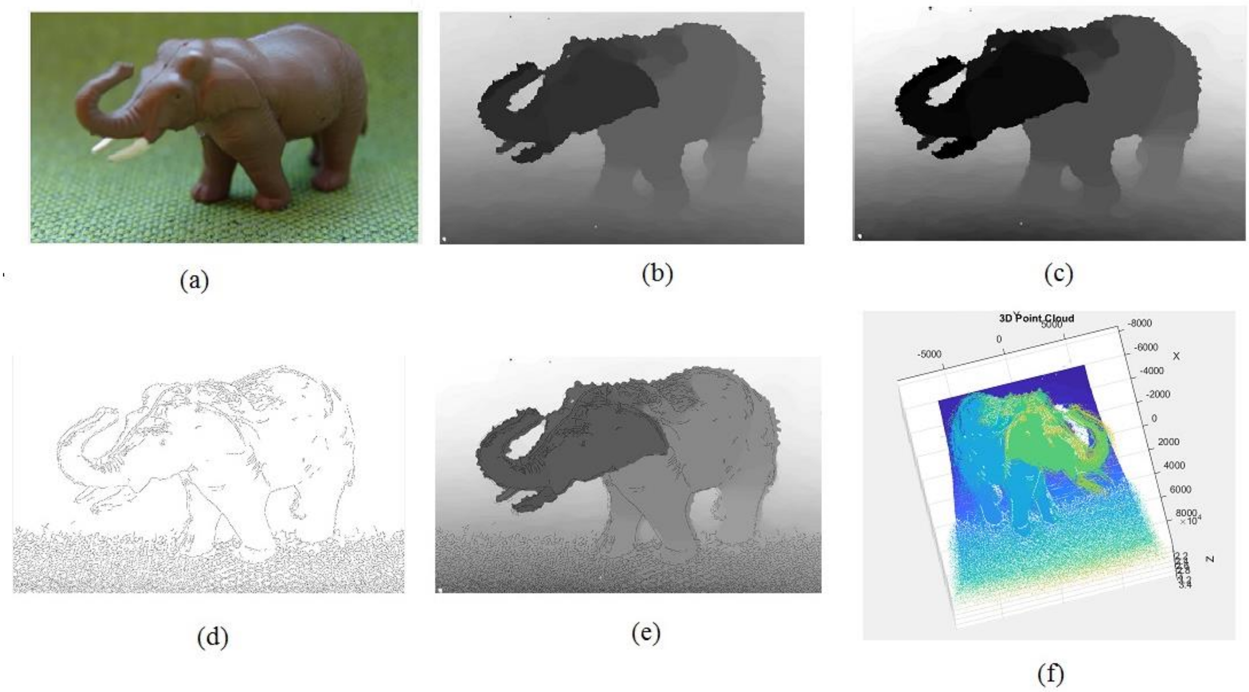

Figure 3. A step by step illustration of our methodology for creating 3D point clouds. (a) Input light field image. (b) Depth map based on our approach. (c) Result of histogram stretching applied to the depth map. (d) The results of applying canny edge detection to the central sub-aperture image. (e) Image fusion (combining (c) and (d) linearly). (f) 3D point cloud.

5.2 Methods compared

To illustrate the accuracy of our proposed method, we compared our result with those Perra et al. [2] and Dansereau et al. [21]. Perra et al. [2], produced 3D point clouds based on the depth map that is created by software supplied with the Lytro camera, however this software does not produce an accurate depth map in all situations. As shown in Figure 4, we compared the depth map output by the Lytro software with our depth map. We re-implemented Perra et al's method in Matlab to compare its performance with our result. For Dansereau et al. [21] we also re-implemented just the 3D reconstruction part of their method in Matlab and evaluated their result against ours. The comparison on a very challenging image is shown in Figure 4. The image is captured in low-light conditions and also has shadowed areas. However, it is clear that our result is more accurate compared to other methods.

5.3 Evaluation methods

For evaluating the performance of point cloud reconstruction algorithms, one important factor is the level of details (LoD) [22]. In computer graphic the level of details is evaluated by number of surface. We calculated the number of points in our point cloud for comparison against two other methods. Table 1 shows the result of this comparison.

Table 1. Comparing numbers of point in synthetic (Papillon) and real image (Cat).

Light field image	**Perra et al.[2]**	**Dansereau et al.[21].**	**Our method**
Papillon	57860	43650	**69523**
Cat	86230	74253	**102365**

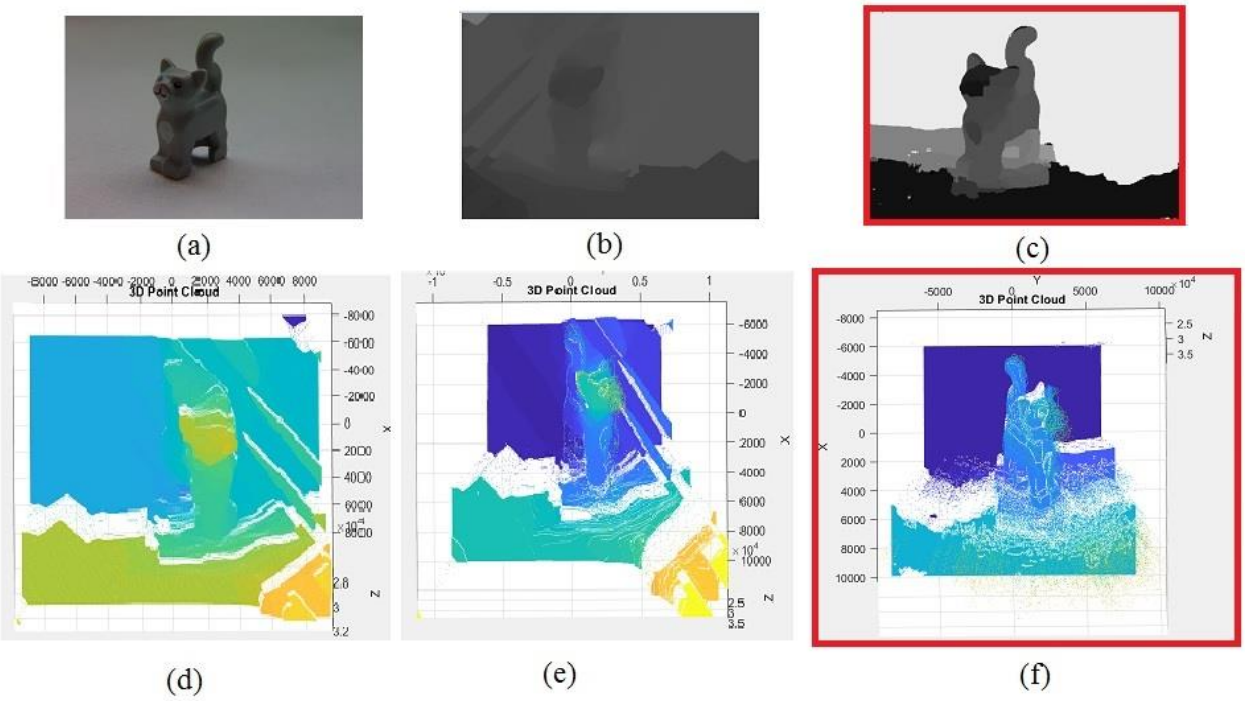

Figure 4. Comparing our result with other state-of-the-art methods. (a) Light field input image; (b) Depth map created by the Lytro software; (c) Depth map that is produced by our method; (d) 3D point cloud obtained by re-implementing the method of Dansereau et al. [21]; (e) 3D point cloud obtained by re-implementing the method of Perra et al. [2]; (f) The result of our 3D point cloud reconstruction algorithm.

CONCLUSION

We have developed a solution for creating 3D point clouds based on the one single image capture. We used a light field image as an input to our system. The unique features of light field cameras lead to accurate depth map estimates. In particular, rich information about scene can be obtained from the one image capture, including light intensity and the direction of light at a range of angles incident to the sensor. In our method, the 3D point cloud has been produced by transformation of the point-plane correspondences. We first estimate the depth map based on the sub aperture image matching, and then we create the 3D point cloud with transformation of the point-plane based on the enhanced depth map. The results confirm that our method can create point clouds with improved accuracy compared to other state-of-the-art methods and our depth map is more accurate than that estimated by the Lytro software. In future work, we will work on converting 3D point cloud to recover the surface geometry of objects for further manipulation.

ACKNOWLEDGEMENTS

This research was supported by a scholarship from Animal Logic Pty. Ltd. JK was supported by an Australian Research Council (ARC) Future Fellowship (FT140100535).

REFERENCES

[1] Yang, B., Rosa, S., Markham, A., Trigoni, N. and Wen, H., "Dense 3D object reconstruction from a single depth view," IEEE transactions on pattern analysis and machine intelligence, (2018).

[2] Perra, C., Murgia, F. and Giusto, D., "An analysis of 3D point cloud reconstruction from light field images," in 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1-6, (2016).

[3] Li, K., Pham, T., Zhan, H. and Reid, I., "Efficient dense point cloud object reconstruction using deformation vector fields," in Proceedings of the European Conference on Computer Vision (ECCV), pp. 497-513, (2018).

[4] Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M. and Hanrahan, P., "Light field photography with a hand-held plenoptic camera," Computer Science Technical Report CSTR, vol. 2, pp. 1-11, (2005).

[5] Pezzuolo, A., Giora, D., Sartori, L. and Guercini, S., "Automated 3D reconstruction of rural buildings from structure-from-motion (SfM) photogrammetry approach," in proceedings of the international scientific conference.[Latvijas Lauksaimniec i¯ bas universit a¯ te], (2018).

[6] Bae, H., White, J., Golparvar-Fard, M., Pan, Y. and Sun, Y., "Fast and scalable 3D cyber-physical modeling for high-precision mobile augmented reality systems," Personal and Ubiquitous Computing, vol. 19, pp. 1275-1294, (2015).

[7] Dimitrov, A. and Golparvar-Fard, M., "Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections," Advanced Engineering Informatics, vol. 28, pp. 37-49, (2014).

[8] Kim, C., Kim, B. and Kim, H. "4D CAD model updating using image processing-based construction progress monitoring," Automation in Construction, vol. 35, pp. 44-52, (2013).

[9] Kim, P., Chen, J. and Cho, Y. K., "SLAM-driven robotic mapping and registration of 3D point clouds," Automation in Construction, vol. 89, pp. 38-48, (2018).

[10] Xia, Y., Wang, C., Xu, Y., Zang, Y., Liu, W., Li, J., et al., "RealPoint3D: Generating 3D Point Clouds from a Single Image of Complex Scenarios," Remote Sensing, vol. 11, p. 2644, (2019).

[11] Yang, G., Huang, X., Hao, Z., Liu, M.-Y., Belongie, S. and B. Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," in Proceedings of the IEEE International Conference on Computer Vision, pp. 4541-4550, (2019).

[12] Wang, L., Meng, W., Xi, R., Zhang, Y., Ma, C., Lu, L., et al., "3D Point Cloud Analysis and Classification in Large-Scale Scene Based on Deep Learning," IEEE Access, vol. 7, pp. 55649-55658, (2019).

[13] Vetrivel, A., Gerke, M., Kerle, N., Nex, F. and G. Vosselman, "Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning," ISPRS journal of photogrammetry and remote sensing, vol. 140, pp. 45-59, (2018).

[14] Wang, W., Yu, R., Huang, Q. and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2569-2578, (2018).

[15] Mandikal, P., Murthy, N., Agarwal, M. and R. V. Babu, "3D-LMNet: Latent Embedding Matching for Accurate and Diverse 3D Point Cloud Reconstruction from a Single Image," arXiv preprint arXiv:1807.07796, (2018).

[16] Hong, D., Yokoya, N., Chanussot, J. and X. X. Zhu, "Cospace: Common subspace learning from hyperspectral-multispectral correspondences," IEEE Transactions on Geoscience and Remote Sensing, (2019).

[17] Wu, G., Masia, B., Jarabo, A., Zhang, Y., Wang, L., Dai, Q., et al., "Light field image processing: An overview," IEEE Journal of Selected Topics in Signal Processing, vol. 11, pp. 926-954, (2017).

[18] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," International journal of computer vision, vol. 60, pp. 91-110, (2004).

[19] Deng, C.-X., Wang, G.-B. and X.-R. Yang, "Image edge detection algorithm based on improved canny operator," in 2013 International Conference on Wavelet Analysis and Pattern Recognition, pp. 168-172, (2013).

[20] Wanner, S., Meister, S. and B. Goldluecke, "Datasets and benchmarks for densely sampled 4D light fields," in VMV, pp. 225-226, (2013).

[21] Dansereau, D. G., Mahon, I., Pizarro, O. and S. B. Williams, "Plenoptic flow: Closed-form visual odometry for light field cameras," in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4455-4462, (2011).

[22] Chen, J., Fang, Y. and Y. K. Cho, "Performance evaluation of 3D descriptors for object recognition in construction applications," Automation in Construction, vol. 86, pp. 44-52, (2018).